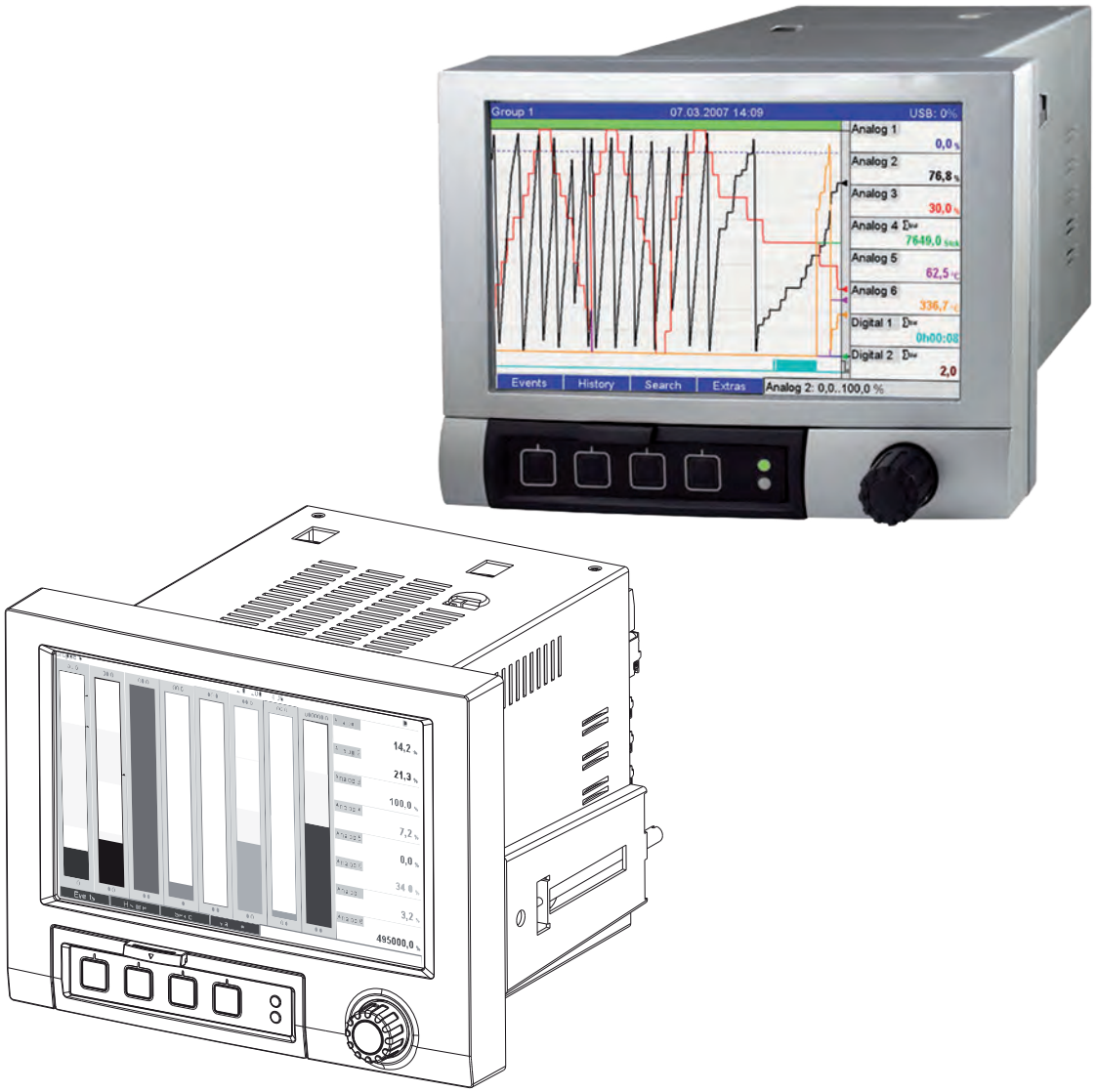
fill level | water level | pressure | temperature | flow | visualization | signal converter | sensoric

Operating Instructions

Regicont RCD-300 Modbus Slave
Advanced Graphic Data Manager



ACS-CONTROL-SYSTEM
knowhow with system
Your partner for measuring technology and automation

Table of contents:

1 General information

Please note the following pictograms:

Note: Suggestions for safe commissioning

Caution: Failure to observe instructions can cause damage to the device or lead to malfunction!

1.1 Requirements

The Modbus module can only be used as of device firmware version V1.02.00 in conjunction with PC software version 1.23.1.0 and higher.

The maths channels 9 to 12 are only supported as of device firmware version V1.10.00 Option „Energy“.

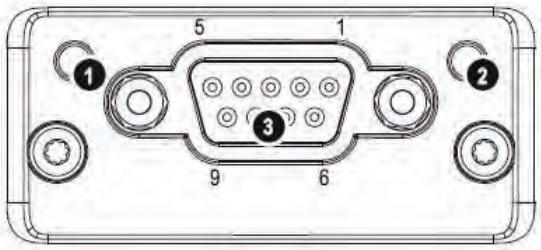
1.2 Scope of delivery

Device with integrated Modbus module.

For more technical information and documentation on your product defined by its serial number please contact your supplier.

1.3 Modbus RTU plug-in module

1.3.1 Connections

1	Communication LED	
2	Status LED	
3	Modbus connector DB9F	

Tab. 1: View of the rear Modbus RTU device connection

1.3.2 Communication LED

Communication LED	Description
Off	Not online / No power Online and data transfer stopped
Flashing yellow (data pulse)	Data transfer active

Tab. 2: Functional description of the communication LED in Modbus RTU

1.3.3 Status LED

Status LED	Description
Off	No power or not initialized
Green	Initialized, no errors
Red	Internal error
Flashing red (1 flash)	Transmission or configuration error
Flashing red (2 flashes)	Diagnosis available

Tab. 3: Functional description of the status LED in Modbus RTU

1.3.4 Modbus RTU connector (DB9F)

The Modbus connector is galvanically isolated and supports RS-232 or RS-485

⚠ Connections are not assigned in the standard way
(Modbus over serial line specification an implementation guide V1.02).

Pin	Direction	Signal	Description
Housing	-	Functional earth	Protective earth
1	-	GND	Earth (isolated)
2	Output[1]	5V	+5V DC (isolated)
3	Input	PMC	Connect to pin 2 for RS-232 functionality. For RS-485 functionality, do not connect.
4	-	-	-
5	Bidirectional	B-Line	RS-485 B-Line
6	-	-	-
7	Input	Rx	RS-232 Data Receive
8	OUTPUT	Tx	RS-232 Data Transmit
9	Bidirectional	A-Line	RS-485 A-Line

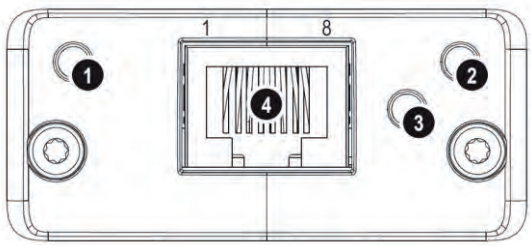
Tab. 4: Pin assignment of the Modbus RTU connector

[1] Any current drawn from this pin will affect the total power consumption of the module.

1.4 Modbus TCP plug-in module

1.4.1 Connections

1	Network status LED
2	Status LED
3	Link/Activity
4	Modbus connector RJ45



Tab. 5: View of the rear Modbus TCP device connection

1.4.2 Network status LED

Note: A test sequence is displayed when the unit is powered up.

Network status LED	Indicates
Off	No power or IP address
Green	Module active
Red	Serious error
Flashing red	Data transfer stopped or no connection
Flashing green	At first initialization and while waiting for connection

Tab. 6: Functional description of the operation mode LED in Modbus TCP

1.4.3 Status LED

Status LED	Indicates
Off	No power or not initialized
Green	Initialized
Flashing red	Initialized, diagnosis available
Red	Exception error

Tab. 7: Functional description of the status LED in Modbus TCP

1.4.4 Link LED

Status LED	Indicates
Off	No connection, no activity
Flashing green	Activity

Tab. 8: Functional description of the link LED in Modbus TCP

1.5 Functional description

The Modbus RTU module allows the device to be connected to Modbus RTU, with the functionality of an RTU slave.
Baud rates supported in baud: 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200

The Modbus TCP module allows the device to be connected to Modbus TCP, with the functionality of a TCP slave. The Ethernet connection supports 10/100Mbit, full or half duplex.

1.6 Checking whether the Modbus module is present

Under **/Main menu/Diagnosis/simulation/Device information/ENP/Hardware**, you can use the **Bus interface** function to check whether a Modbus module is used. The software version and serial number are visible here, and for Modbus TCP, the MAC address also.

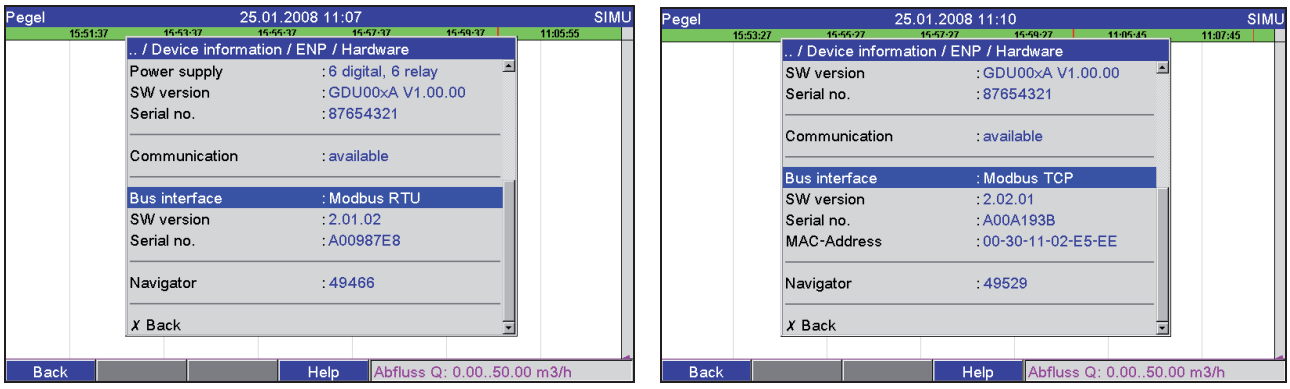


Fig. 1: Checking whether the Modbus module is present

2 Settings in the Setup

Modbus RTU:

A slave address between 1 and 247 is configured under **/Setup/System/Modbus** (see Fig. 2).

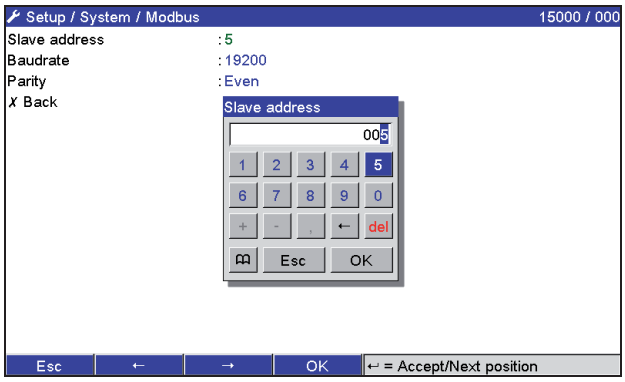


Fig. 2: Entering the slave address in Modbus RTU

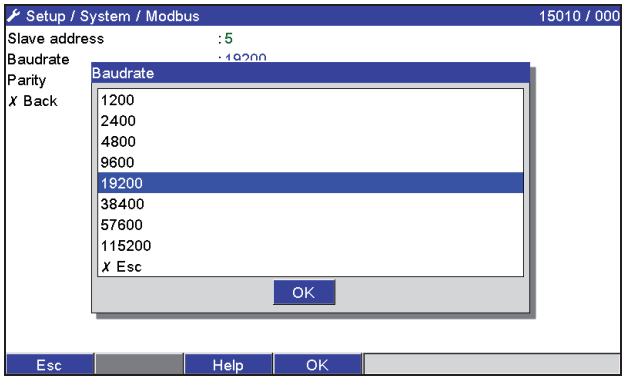


Fig. 3: Entering the baudrate in Modbus RTU

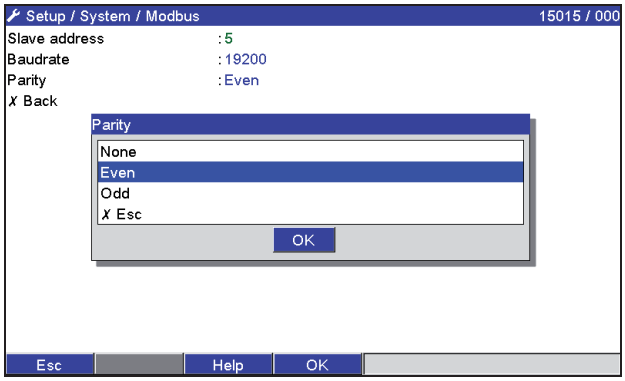


Fig. 4: Selecting the parity in Modbus RTU

Fig. 5: Selecting the timeout in Modbus RTU

Modbus TCP:

The IP address is configured under **/Setup/System/Modbus** (see Fig. 6 to Fig. 8). You can choose between DHCP and manual entry:

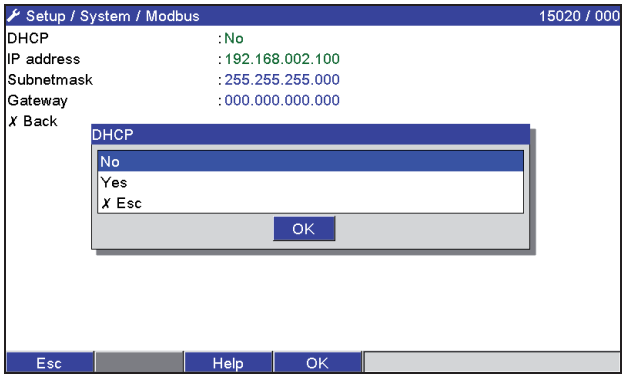


Fig. 6: Selecting DHCP in Modbus TCP

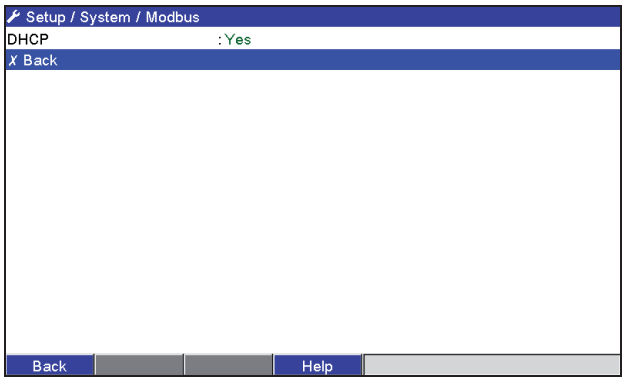


Fig. 7: Automatic IP address assignment in Modbus TCP

If the IP is entered manually, IP address, subnetmask and gateway must be entered (see Fig. 8):

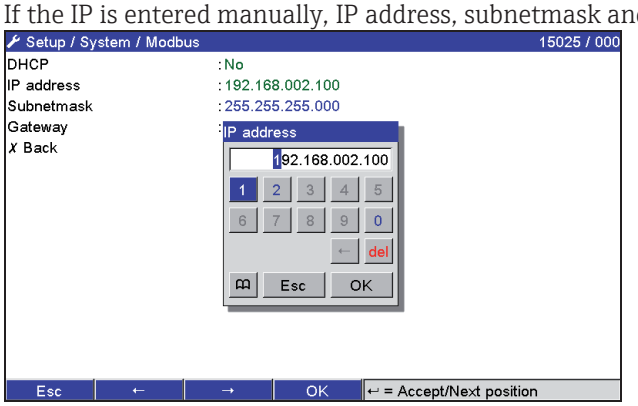


Fig. 8: Selecting manual entry of IP address in Modbus TCP

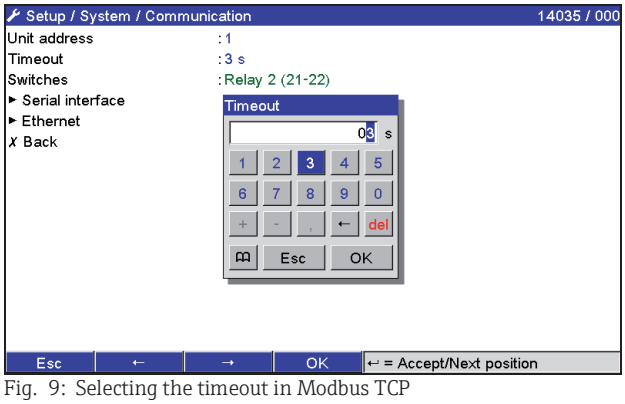


Fig. 9: Selecting the timeout in Modbus TCP

The IP address assigned using DHCP can be viewed under /**Main menu/Diagnosis/Simulation/Device information/ENP**.

2.1 Analog channels

All analog (40) and digital (14) inputs are enabled and can be used as Modbus inputs even if they are not really available as plug-in cards.

Data transfer Modbus master -> Device:
Under **/Setup/Inputs/Analog inputs/Analog input X**, the **Signal** parameter is set to **Modbus**.
The analog channel configured in this way can be selected for data transfer (see Section 3.2.1).

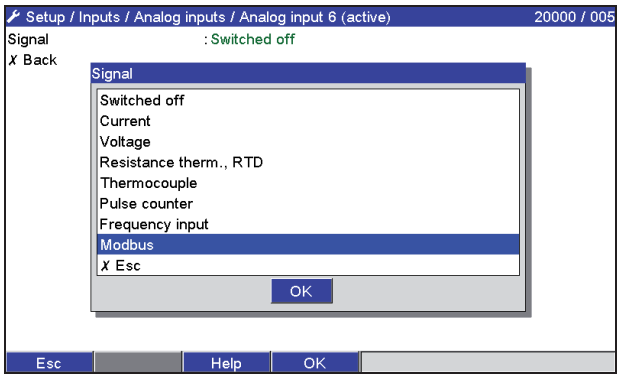


Fig. 10: Setting the analog channel to Modbus

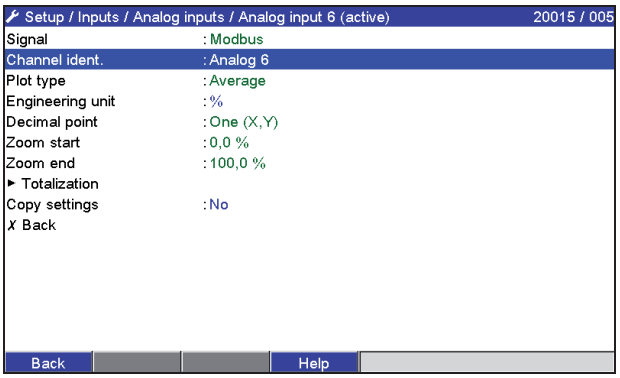


Fig. 11: Selecting the desired channel

Data transfer, device -> Modbus master
Analog inputs 1 to 20 can be read by the Modbus master as described in Section 3.2.1.

2.2 Mathematics channels

Data transfer, device -> Modbus master
Mathematics channels are optionally available under **/Setup/Inputs/Maths**.
The results can be read by the Modbus master (see Section 3.2.4).

2.3 Digital channels

Data transfer, Modbus master -> Device:
Under **/Setup/Inputs/Digital inputs/Digital input X**, the **Function** parameter is set to **Modbus**.
The digital channel configured in this way can be used for data transfer (see Section 3.2.2).

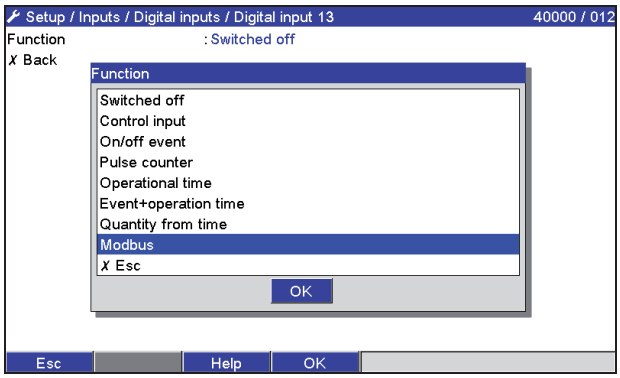


Fig. 12: Setting the digital channel to Modbus

The digital status transmitted by the Modbus master has the same functionality in the device as the status of a digital channel really available.

Data transfer, device -> Modbus master
Control input or on/off event
The digital status of the digital channel configured in this way can be read by the Modbus master (see Section 3.2.5).

Pulse counter or operating time
The counter or the total operating time of the digital channel configured in this way can be read by the Modbus master (see Section 3.2.5).

Event+operation time
The digital status and counter of the digital channel configured in this way can be read by the Modbus master (see Section 3.2.5).

An overview can be called up via a Web browser (Ethernet option). The IP address to be specified is that of the device and not the Modbus module (TCP).

E.g. URL: http://192.168.100.7/fieldbus

ModbusETH V3.03.01 A017E7DB
MAC: 00-30-11-07-6D-5B
Link active, IP established, DHCP not active,
IP: 192.168.178.200 SM: 255.255.255.0 GW: 0.0.0.

Write Multiple Register (16)

Reg.	Channel	Reg.	Channel
0	Analog 1	60	Analog 21
3	Analog 2	63	Analog 22
6	Analog 3	66	Analog 23
9	Analog 4	69	Analog 24
12	Analog 5	72	Analog 25
15	Analog 6	75	Analog 26
18	Analog 7	78	Analog 27
21	Analog 8	81	Analog 28
24	Analog 9	84	Analog 29
27	Analog 10	87	Analog 30
30	Analog 11	90	Analog 31
33	Analog 12	93	Analog 32
36	Analog 13	96	Analog 33
39	Analog 14	99	Analog 34
42	Analog 15	102	Analog 35
45	Analog 16	105	Analog 36
48	Analog 17	108	Analog 37
51	Analog 18	111	Analog 38
54	Analog 19	114	Analog 39
57	Analog 20	117	Analog 40
120	Digital 1-14		

Fig. 13: Web site of Modbus overview

Read Holding Register (03)

Reg.	Channel	Reg.	Channel
256	Analog 1	316	Mathe 1
259	Analog 2	319	Mathe 2
262	Analog 3	322	Mathe 3
265	Analog 4	325	Mathe 4
268	Analog 5	328	Mathe 5
271	Analog 6	331	Mathe 6
274	Analog 7	334	Mathe 7
277	Analog 8	337	Mathe 8
280	Analog 9	736	Mathe 9
283	Analog 10	740	Mathe 10
286	Analog 11	744	Mathe 11
289	Analog 12	748	Mathe 12
292	Analog 13	340	Digital 1
295	Analog 14	343	Digital 2
298	Analog 15	346	Digital 3
301	Analog 16	349	Digital 4
304	Analog 17	352	Digital 5
307	Analog 18	355	Digital 6
310	Analog 19	358	Digital 7
313	Analog 20	361	Digital 8
784	Analog 21	364	Digital 9
788	Analog 22	367	Digital 10
792	Analog 23	370	Digital 11
796	Analog 24	373	Digital 12
800	Analog 25	376	Digital 13
804	Analog 26	379	Digital 14
808	Analog 27		
812	Analog 28		
816	Analog 29		
820	Analog 30		
824	Analog 31		
828	Analog 32		
832	Analog 33		
836	Analog 34		
840	Analog 35		
844	Analog 36		
848	Analog 37		
852	Analog 38		
856	Analog 39		
860	Analog 40		

3 Data transmission

3.1 General information

The **03: Read Holding Register** and **16: Write Multiple Register** functions are supported.

You can transfer
- Analog values (instantaneous values)
- Digital statuses
- Text

from the **Modbus master to the device**.

You can transfer
- Analog values (instantaneous values)
- Integrated analog values (counter)
- Mathematics channels (result: status, instantaneous value, operating time, counter)
- Integrated maths channels (counter)
- Digital status
- Pulse counter (overall counter)
- Operating times

from the **device to the Modbus master**.

3.2 Addressing

The query/response samples refer to Modbus RTU.
The register addresses are all on base 0.

3.2.1 Modbus master -> Device: analog channels instantaneous value

The values of analog channels 1-40 must be written via **16 Write Multiple Register**.

Channel	Reg. Dec.	Reg. Hex.	Length in bytes
Analog 1	0	000	6
Analog 2	3	003	6
Analog 3	6	006	6
Analog 4	9	009	6
Analog 5	12	00C	6
Analog 6	15	00F	6
Analog 7	18	012	6
Analog 8	21	015	6
Analog 9	24	018	6
Analog 10	27	01B	6
Analog 11	30	01E	6
Analog 12	33	021	6
Analog 13	36	024	6
Analog 14	39	027	6
Analog 15	42	02A	6
Analog 16	45	02D	6
Analog 17	48	030	6
Analog 18	51	033	6
Analog 19	54	036	6
Analog 20	57	039	6
Analog 21	60	03C	6
Analog 22	63	03F	6
Analog 23	66	042	6
Analog 24	69	045	6
Analog 25	72	048	6
Analog 26	75	04B	6
Analog 27	78	04E	6
Analog 28	81	051	6
Analog 29	84	054	6
Analog 30	87	057	6
Analog 31	90	05A	6
Analog 32	93	05D	6
Analog 33	96	060	6
Analog 34	99	063	6
Analog 35	102	066	6
Analog 36	105	069	6
Analog 37	108	06C	6
Analog 38	111	06F	6
Analog 39	114	072	6
Analog 40	117	075	6

Tab. 9: Register addresses of the analog inputs, Modbus master -> Device

The status of the floating point number transmitted in the 2nd and 3rd register is found in the first register (see Section 3.2.12.3).

Example: **Writing to analog 17, value 123.456, slave address 5**

Byte	0	1	2	3	4	5
	00	80	42	F6	E9	79
		Status floating point number	Floating point number = 123.456			

Register	Value (hex)
48	00**80**
49	**42F6**
50	**E979**

Query:			
	Slave address	05	
	Function	10	16: Write Multiple Registers
	Register	00 30	Register 48
	No. of registers	00 03	3 registers
	No. of bytes	06	
	Status	00 80	
	FLP	42 F6 E9 79	123.456
	CRC	93 1D	

Response:			
	Slave address	05	
	Function	10	16: Write Multiple Registers
	Register	00 30	Register 48
	No. of registers	00 03	
	CRC	81 33	

3.2.2 Modbus master -> Device: digital input status

The statuses of analog inputs 1-14 must be written via **16 Write Multiple Register**.

Channel	Reg. Dec.	Reg. Hex.	Length in bytes
Digital 1-14	120	078	4

Tab. 10: Register addresses of the digital inputs, Modbus master -> Device

The new statuses of the analog inputs are found in the first register (120). The mask which describes if the status is adopted is found in the second register (121).

Example: **Setting digital input 8 to High and digital input 9 to Low, slave address 5**

Byte 0 Status (Bit 15-8)	Byte 1 Status (Bit 7-0)	Byte 2 Mask (Bit 15-8)	Byte 3 Mask (Bit 7-0)
XX000000	10000000	XX000001	10000000
Bit 8 Low Digital 9	Bit 7 High Digital 8	Bit 8 High Digital 9 active	Bit 7 High Digital 8 active

Register	Value (hex)
120	0080
121	0180

Query:	Slave address	05	
	Function	10	16: Write Multiple Registers
	Register	00 78	Register 120
	No. of registers	00 02	2 registers
	No. of bytes	04	
	Digital status	00 80	Digital 8 set to High, Digital 9 set to Low
	Mask	01 80	Digital 8 and 9 masked
	CRC	E1 C5	
Response:	Slave address	05	
	Function	10	16: Write Multiple Registers
	Register	00 78	Register 120
	No. of registers	00 02	
	CRC	C0 55	

3.2.3 Device -> Modbus master: analog inputs instantaneous value

The analog channels 1-20 are read via **03 Read Holding Register (4x)**.

Channel	Reg. Dec.	Reg. Hex.	Length in bytes
Analog 1	256	100	6
Analog 2	259	103	6
Analog 3	262	106	6
Analog 4	265	109	6
Analog 5	268	10C	6
Analog 6	271	10F	6
Analog 7	274	112	6
Analog 8	277	115	6
Analog 9	280	118	6
Analog 10	283	11B	6

Channel	Reg. Dec.	Reg. Hex.	Length in bytes
Analog 11	286	11E	6
Analog 12	289	121	6
Analog 13	292	124	6
Analog 14	295	127	6
Analog 15	298	12A	6
Analog 16	201	12D	6
Analog 17	304	130	6
Analog 18	307	133	6
Analog 19	310	136	6
Analog 20	313	139	6

Tab. 11: Register addresses of the analog inputs, device -> Modbus master

Channel	Reg. Dec.	Reg. Hex.	Length in bytes
Analog 21	784	310	6
Analog 22	788	314	6
Analog 23	792	318	6
Analog 24	796	31C	6
Analog 25	800	320	6
Analog 26	804	324	6
Analog 27	808	328	6
Analog 28	812	32C	6
Analog 29	816	330	6
Analog 30	820	334	6

Channel	Reg. Dec.	Reg. Hex.	Length in bytes
Analog 31	824	338	6
Analog 32	828	33C	6
Analog 33	832	340	6
Analog 34	836	344	6
Analog 35	840	348	6
Analog 36	844	34C	6
Analog 37	848	350	6
Analog 38	852	354	6
Analog 39	856	358	6
Analog 40	860	35C	6

Tab. 181a: Advanced register addresses of the analog inputs, device -> Modbus master

The status of the floating point number transmitted in the 2nd and 3rd register is found in the first register (see Section 3.2.12.3).

Example: **Reading analog 2 , value 5.016928673, slave address 5**

Byte	0	1	2	3	4	5
	00	80	42	2C	1F	BA
	Off-limit conditions	Status floating point number	Floating point number = 43.030983			

Register	Value (hex)
259	0080
260	422C
261	1FBA

Query:	Slave address	05	
	Function	03	03: Read Holding Register
	Register	01 03	Register 259
	No. of registers	00 03	3 registers
	CRC	F5 B3	
Response:	Slave address	05	
	Function	03	03: Read Holding Register
	No. of bytes	06	6 bytes
	Status	00 80	
	FLP	42 2C 1F BA	43.030983
	CRC	4E 59	

3.2.4 Device -> Modbus master: maths channels result

The results of the mathematics channels are read via **03 Read Holding Register (4x)**.

Channel	Reg. Dec.	Reg. Hex.	Length in bytes
Maths 1	316	13C	6
Maths 2	319	13F	6
Maths 3	322	142	6
Maths 4	325	145	6
Maths 5	328	148	6
Maths 6	331	14B	6
Maths 7	334	14E	6
Maths 8	337	151	6
Mathe 9	736	2E0	6
Mathe 10	740	2E4	6
Mathe 11	744	2E8	6
Mathe 12	748	2EC	6

Tab. 12: Register addresses of the mathematics channels, device -> Modbus master

The status of the floating point number transmitted in the 2nd and 3rd register is found in the first register (see Section 3.2.12.3).

Example: **Reading maths 1 (result instantaneous value), slave address 5**

Byte	0	1	2	3	4	5
	00	80	41	A0	00	00
	Digital status / Off-limit conditions	Status floating point number	Floating point number = 20.0			

Register	Value (hex)
316	00**80**
317	**41A0**
318	**0000**

Query:	Slave address	05	
	Function	03	03: Read Holding Register
	Register	01 3C	Register 316
	No. of registers	00 03	3 registers
	CRC	C5 BF	
Response:	Slave address	05	
	Function	03	03: Read Holding Register
	No. of bytes	06	6 bytes
	Status	00 80	
	FLP	41 A0 00 00	20.0
	CRC	06 75	

Example: **Reading maths 1 (result status), slave address 5**

The status is found in the first register, high byte.

Byte	0	1	2	3	4	5
	01	00	00	00	00	00
	Digital status					

Register	Value (hex)
316	**01**00
317	0000
318	0000

Query:			
	Slave address	05	
	Function	03	03: Read Holding Register
	Register	01 3C	Register 316
	No. of registers	00 03	3 registers
	CRC	C5 BF	

Response:			
	Slave address	05	
	Function	03	03: Read Holding Register
	No. of bytes	06	6 bytes
	Status	01	Result maths 1 = High
		00 00 00 00 00	Not used
	CRC	12 64	

3.2.5 Device -> Modbus master: digital channels (status, pulse counter)

The statuses and values of the pulse counter (total counter) are read via **03 Read Holding Register (4x)**.

Channel	Reg. Dec.	Reg. Hex.	Length in bytes
Digital 1	340	154	6
Digital 2	343	157	6
Digital 3	346	15A	6
Digital 4	349	15D	6
Digital 5	352	160	6
Digital 6	355	163	6
Digital 7	358	166	6

Channel	Reg. Dec.	Reg. Hex.	Length in bytes
Digital 8	361	169	6
Digital 9	364	16C	6
Digital 10	367	16F	6
Digital 11	370	172	6
Digital 12	373	175	6
Digital 13	376	178	6
Digital 14	379	17B	6

Tab. 13: Register addresses of the digital channels, device -> Modbus master

The status of the floating point number transmitted in the 2nd and 3rd register is found in the first (low byte) register (see Section 3.2.12.3).
The digital status is found in the first register (high byte, bit 0).

Example: **Reading digital 2 (status), slave address 5**

The status is found in the first register, high byte.

Byte	0	1	2	3	4	5
	01	00	00	00	00	00
	Digital status					

Register	Value (hex)
343	**01**00
344	0000
345	0000

Query:	Slave address	05	
	Function	03	03: Read Holding Register
	Register	01 57	Register 343
	No. of registers	00 03	3 registers
	CRC	B4 63	
Response:	Slave address	05	
	Function	03	03: Read Holding Register
	No. of bytes	06	6 bytes
	Status	01	Result digital = High
		00 00 00 00 00	Not used
	CRC	12 64	

Example: **Reading digital 2 (pulse counter), slave address 5**

Byte	0	1	2	3	4	5
	00	80	40	A0	00	00
	Digital status	Status floating point number	Floating point number = 5.0			

Register	Value (hex)
343	00**80**
344	**40A0**
345	**0000**

Query:	Slave address	05	
	Function	03	03: Read Holding Register
	Register	01 57	Register 343
	No. of registers	00 03	3 registers
	CRC	B4 63	
Response:	Slave address	05	
	Function	03	03: Read Holding Register
	No. of bytes	06	6 bytes
	Status	01	Result digital = High
	Status	80	
	FLP	40 A0 00 00	Pulse counter to 5.0
	CRC	06 58	

3.2.6 Device -> Modbus master: integrated analog channels (counter)

The integrated values of analog inputs 1-40 are read via **03 Read Holding Register (4x)**.

Channel	Reg. Dec.	Reg. Hex.	Length in bytes
Analog 1	528	210	6
Analog 2	532	214	6
Analog 3	536	218	6
Analog 4	540	21C	6
Analog 5	544	220	6
Analog 6	548	224	6
Analog 7	552	228	6
Analog 8	556	22C	6
Analog 9	560	230	6
Analog 10	564	234	6
Analog 11	568	238	6
Analog 12	572	23C	6
Analog 13	576	240	6
Analog 14	580	244	6
Analog 15	584	248	6
Analog 16	588	24C	6
Analog 17	592	250	6
Analog 18	596	254	6
Analog 19	600	258	6
Analog 20	604	25C	6

Channel	Reg. Dec.	Reg. Hex.	Length in bytes
Analog 21	608	260	6
Analog 22	612	264	6
Analog 23	616	268	6
Analog 24	620	26C	6
Analog 25	624	270	6
Analog 26	628	274	6
Analog 27	632	278	6
Analog 28	636	27C	6
Analog 29	640	280	6
Analog 30	644	284	6
Analog 31	648	288	6
Analog 32	652	28C	6
Analog 33	656	290	6
Analog 34	660	294	6
Analog 35	664	298	6
Analog 36	668	29C	6
Analog 37	672	2A0	6
Analog 38	676	2A4	6
Analog 39	680	2A8	6
Analog 40	684	2AC	6

Tab. 14: Register addresses of the integrated analog inputs, device -> Modbus master

The status of the floating-point number transmitted in the 2nd and 3rd register is found in the first (low byte) register (see Section 3.2.12.3).

Example: **Reading the counter of integrated analog input 5**

Byte	0	1	2	3	4	5
	00	80	43	E8	46	BB
	Off-limit conditions	Status floating point number	Floating point number = 464.55			

Register	Value (hex)
544	0080
545	43E8
546	D417

Query:	Slave address	05	
	Function	03	03: Read Holding Register
	Register	02 20	Register 544
	No. of registers	00 03	3 registers
	CRC	04 3D	
Response:	Slave address	05	
	Function	03	03: Read Holding Register
	No. of bytes	06	6 bytes
	Status	00 80	
	FLP	43 E8 46 BB	Integrated value to 464.55
	CRC	F5 C8	

3.2.7 Device -> Modbus master: integrated maths channels (counter)

The integrated values of mathematics channels 1-12 are read via **03 Read Holding Register (4x)**.

Channel	Reg. Dec.	Reg. Hex.	Length in bytes
Maths 1	688	2B0	6
Maths 2	692	2B4	6
Maths 3	696	2B8	6
Maths 4	700	2BC	6
Maths 5	704	2C0	6
Maths 6	708	2C4	6
Maths 7	712	2C8	6
Maths 8	716	2CC	6
Mathe 9	720	2D0	6
Mathe 10	724	2D4	6
Mathe 11	728	2D8	6
Mathe 12	732	2DC	6

Tab. 15: Register addresses of the integrated maths channels, device -> Modbus master

The status of the floating-point number transmitted in the 2nd and 3rd register is found in the first (low byte) register (see Section 3.2.12.3).

Example: **Reading the counter of integrated mathematics channel 1**

Byte	0	1	2	3	4	5
	00	80	45	1D	C0	00
	Off-limit conditions	Status floating point number	Floating point number = 2524			

Register	Value (hex)
688	00**80**
689	**451D**
690	**C000**

Query:	Slave address	05	
	Function	03	03: Read Holding Register
	Register	02 B0	Register 688
	No. of registers	00 03	3 registers
	CRC	04 10	
Response:	Slave address	05	
	Function	03	03: Read Holding Register
	No. of bytes	06	6 bytes
	Status	00 80	
	FLP	45 1D C0 00	Integrated value to 2524
	CRC	C7 61	

3.2.8 Modbus master -> Device: batch data

Firmware version V2.00.00 and higher

Batches can be started and stopped. The batch name, batch designation, batch number and preset counter can also be configured for stopping the batch. The texts (ASCII) can have a maximum length of 30 characters. If the text is longer than 30 characters it is truncated and saved.
The functions and texts must be written using **16 Write Multiple Register**.
If an uneven number of characters is sent, a blank (0x20) must follow. The blank is ignored in the device.

Function	Description	Data
0x01	Start batch	Batch (1 to 4), ID, name
0x02	Stop batch	Batch (1 to 4), ID, name
0x03	Batch designation	Batch (1 to 4), text (max 30 characters)
0x04	Batch name	Batch (1 to 4), text (max 30 characters)
0x05	Batch number	Batch (1 to 4), text (max 30 characters)
0x06	Preset counter	Batch (1 to 4), text (max 12 characters)

3.2.8.1 Starting a batch

If the user administration function is enabled, an ID (max. 8 characters) and a name (max. 20 characters) must be transmitted, separated by a ';'. If an uneven number of characters is sent, a blank (0x20) must follow (see 3.2.8.2 Ending a batch).

Example: **Start batch 2 (without user administration function)**

Byte	0	1
	func	no.
	1	2

Register	Value (hex)
3088	0102

Query:	Slave address	05	
	Function	10	16: Write Multiple Register
	Register	0C 10	Register 3088
	No. of registers	00 01	1 register
	No. of bytes	02	2 bytes
	Data	01 02	
	CRC	D2 51	
Response:	Slave address	05	
	Function	10	16: Write Multiple Register
	Register	0C 10	Register 3088
	No. of registers	00 01	1 register
	CRC	02 D8	

The entry "Batch 2 started" is saved in the event list. This message also appears on the screen for a few seconds.

3.2.8.2 Ending a batch

If the user administration function is enabled, an ID (max. 8 characters) and a name (max. 20 characters) must be transmitted, separated by a ';'. If an uneven number of characters is sent, a blank (0x20) must follow.

Example: **End batch 2, user administration function enabled (ID: "IDSPS", Name "RemoteX")**

Byte	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	func	no.	49	44	53	50	53	3B	52	65	6D	6F	74	65	58	20
	2	2	'I'	'D'	'S'	'P'	'S'	';'	'R'	'e'	'm'	'o'	't'	'e'	'X'	, '

Register	Value (hex)
3088	0202
3089	4944
3090	5350
3091	533B
3092	5265
3093	6D6F
3094	7465
3095	5820

Query:	Slave address	05	
	Function	10	16: Write Multiple Register
	Register	0C 10	Register 3088
	No. of registers	00 08	8 registers
	No. of bytes	10	16 bytes
	Data	02 02 49 44 53 59 53 3B 52 65 6D 6F 74 65 58 20	
	CRC	D3D6	
Response:	Slave address	05	
	Function	10	16: Write Multiple Register
	Register	0C 10	Register 3088
	No. of registers	00 08	8 registers
	CRC	C2 DE	

The entry "Batch 2 terminated" and "Remote (IDSPS)" is saved in the event list. This message also appears on the screen for a few seconds.

3.2.8.3 Setting the batch designation

Can only be set if the batch has not yet been started. It does not have to be set if it is not required by the device settings (Direct access 16070).

Example: **Batch designation "Identifier" for batch 2**

Byte	0	1	2	3	4	5	6	7	8	9	10	11
	func	no.	49	64	65	6E	74	69	66	69	65	72
	3	2	'I'	'd'	'e'	'n'	't'	'i'	'f'	'i'	'e'	'r'

Register	Value (hex)
3088	0302
3089	5964
3090	656E
3091	7469
3092	6669
3093	6572

Query:

Slave address	05	
Function	10	16: Write Multiple Register
Register	0C 10	Register 3088
No. of registers	00 06	6 registers
No. of bytes	0B	12 byte
Data	03 02 59 64 65 6E 74 69 66 69 65 72	
CRC	0E 20	

Response:

Slave address	05	
Function	10	16: Write Multiple Register
Register	0C 10	Register 3088
No. of registers	00 06	6 registers
CRC	43 1A	

3.2.8.4 Setting the batch name

Can only be set if the batch has not yet been started. It does not have to be set if it is not required by the device settings (Direct access 16071).

Example: **Batch name "Name" for batch 2**

Byte	0	1	2	3	4	5
	func	no.	4E	61	6D	65
	4	2	'N'	'a'	'm'	'e'

Register	Value (hex)
3088	0402
3089	4E61
3090	6D65

Query:	Slave address	05	
	Function	10	16: Write Multiple Register
	Register	0C 10	Register 3088
	No. of registers	00 03	3 registers
	No. of bytes	06	6 bytes
	Data	04 02 4E 61 6D 65	
	CRC	04 C8	
Response:	Slave address	05	
	Function	10	16: Write Multiple Register
	Register	0C 10	Register 3088
	No. of registers	00 03	3 registers
	CRC	83 19	

3.2.8.5 Setting the batch number

Can only be set if the batch has not yet been started. It does not have to be set if it is not required by the device settings (Direct access 16072).

Example: **Batch number "Num" for batch 2**

Byte	0	1	2	3	4	5
	func	no.	4E	75	6D	20
	5	2	'N'	'u'	'm'	, ’

Register	Value (hex)
3088	0502
3089	4E75
3090	6D20

Query:	Slave address	05	
	Function	10	16: Write Multiple Register
	Register	0C 10	Register 3088
	No. of registers	00 03	3 registers
	No. of bytes	06	6 bytes
	Data	05 02 4E 75 6D 20	
	CRC	84 EE	
Response:	Slave address	05	
	Function	10	16: Write Multiple Register
	Register	0C 10	Register 3088
	No. of registers	00 03	3 registers
	CRC	83 19	

3.2.8.6 Setting the preset counter

Can only be set if the batch has not yet been started. It does not have to be set if it is not required by the device settings (Direct access 16073).

- Maximum 12 characters (including '.')
- Exponential function permitted, e.g. "1.23E-2"
- Only positive numbers

Example: **Preset counter to 12.345 for batch 2**

Byte	0	1	2	3	4	5	6	7
	func	no.	31	32	2E	33	34	35
	6	2	'1'	'2'	'.'	'3'	'4'	'5'

Register	Value (hex)
3088	0602
3090	3132
3091	2E33
3092	3435

Query:			
	Slave address	05	
	Function	10	16: Write Multiple Register
	Register	0C 10	Register 3088
	No. of registers	00 04	4 registers
	No. of bytes	08	8 bytes
	Data	06 02 31 32 2E 33 34 35	
	CRC	D3 B5	

Response:			
	Slave address	05	
	Function	10	16: Write Multiple Register
	Register	0C 10	Register 3088
	No. of registers	00 04	4 registers
	CRC	C2 DB	

3.2.8.7 Reading out the batch status

This can be used to read out the status of every batch and the last communication status.

Example: **Batch 2 started, communication status "OK"**

Query:	Slave address	05	
	Function	03	03: Read holding register (4x)
	Register	0C 10	Register 3088
	No. of registers	00 03	3 registers
	CRC	06 DA	
Response:	Slave address	05	
	Function	3	03: Read holding register (4x)
	Register	0C 10	Register 3088
	No. of bytes	6	6 bytes
	Data	00 00 00 01 00 00	
	CRC	42 75	

Byte	0	1	2	3	4	5
		Comm. status	Status batch 1	Status batch 2	Status batch 3	Status batch 4
	0	0	0	1	0	0

Register	Value (hex)
3088	0000
3090	0001
3091	0000

If, for example, a batch number is set even though the batch is already running, the value 0x0003 would be in register 3088.

Communication status:
0: OK
1: Not all the necessary data were transmitted (mandatory entries)
2: User responsible not logged on
3: Batch already running
4: Batch not configured
5: Batch controlled via control input
7: Automatic batch number active
9: Error, text contained characters that cannot be displayed, text too long, incorrect batch number
Function number out of range

Batch status:
0: Batch inactive
1: Batch active

3.2.9 Modbus master -> Device: set relays

Firmware version V2.00.00 and higher

Relays can be set if they were set to "Remote" in the device settings. **16 Write Multiple Register** or **06 Write Single Register** can be used for this purpose.

3.2.9.1 Setting relays

Relay status:
0: Inactive
1: Active

Example: **Setting relay 6 to the active state**

Byte	0	1
	RelNo	Status
	6	1

Register	Value (hex)
3152	0601

Query:	Slave address	05	
	Function	10	16: Write Multiple Register
	Register	0C 50	Register 3152
	No. of registers	00 01	1 register
	No. of bytes	02	2 bytes
	Data	06 01	
	CRC	96 A0	
Response:	Slave address	05	
	Function	10	16: Write Multiple Register
	Register	0C 50	Register 3152
	No. of registers	00 01	1 register
	CRC	03 0C	

3.2.9.2 Reading out the relay status

This reads out the status of every relay. Bit 0 corresponds to relay 1.

Example: **Relay 1 and relay 6 in an active state**

Query:	Slave address	05	
	Function	03	03: Read holding register (4x)
	Register	0C 50	Register 3152
	No. of registers	00 03	3 registers
	CRC	86 CF	
Response:	Slave address	05	
	Function	3	03: Read holding register (4x)
	Register	0C 50	Register 3152
	No. of bytes	2	2 bytes
	Data	00 21	
	CRC	89 9C	

Register	Value (hex)
3152	0021

The relay status is determined from the 2 data bytes as follows:

Byte 0:	Byte 1:
Bit 0 = Status relay 1	Bit 0 = Status relay 9
Bit 1 = Status relay 2	Bit 1 = Status relay 10
Bit 2 = Status relay 3	Bit 2 = Status relay 11
Bit 3 = Status relay 4	Bit 3 = Status relay 12
Bit 4 = Status relay 5	
Bit 5 = Status relay 6	
Bit 6 = Status relay 7	
Bit 7 = Status relay 8	

Example: "0E07" returns the following status of the relays:
Relays 1-3 and relays 10-12 active.

3.2.10 Modbus master -> Device: changing the limit values

Firmware version V2.00.00 and higher

Limit values can be set by using **16 Write Multiple Register** or **06 Write Single Register** for this purpose.

Function	Description	Data
0x01	Initialization	
0x02	Accept limit values	
0x03	Change limit value	Limit value number;value;Time span for gradient;Delay
0x04	Read out limit value	Limit value settings
0x05	Enter reason	Text containing the reason

To change limit values, the following sequence must be observed:
1. Initialize change of limit values
2. Change limit values
3. Accept limit values

In versions earlier than firmware version V2.00.04
A subsequent limit value change cannot be initialized until the limit values have been accepted.
In versions from firmware version V2.00.04
Any changes since the last initialization can be discarded when a subsequent limit value change is initialized.
In versions from firmware version V2.10.00
The delay time can also be configured in addition to the limit value.
The activated limit values can be read out.
In versions from firmware version V2.10.02
A reason for the change in the limit value can be entered.

3.2.10.1 Initializing limit value changes

This prepares the device for changes to the limit values.
16 Write Multiple Register or **06 Write Single Register** can be used for this purpose.

Byte	0	1
	Func	Fill byte
	1	2A

Register	Value (hex)
3216	012A

Query:	Slave address	05	
	Function	10	16: Write Multiple Register
	Register	0C 90	Register 3216
	No. of registers	00 01	1 register
	No. of bytes	02	2 bytes
	Data	01 2A	
	CRC	96 A0	
Response:	Slave address	05	
	Function	10	16: Write Multiple Register
	Register	0C 90	Register 3216
	No. of registers	00 01	1 register
	CRC	03 30	

3.2.10.2 Changing limit values

Here, a limit value in the device is changed but is not yet accepted.
The transmitted values are separated by a semicolon (;).

The following structure must be observed: Func limit value [Value];[Span];[Delay type];[Delay]
[] means that this value can also be omitted. Likewise, only the values that should be changed need to be transmitted.

Examples (see also 3.2.10.7 Tables and definitions**):**

Func	Limit value	Data	Meaning
3	1	5.22;;1;2	Limit value 1 to 5.22, no span, delay in minutes, 2 minutes
3	2	5.34	Limit value 2 to 5.34
3	3	;;0;10	Limit value 3, delay to 10 seconds

If an uneven number of characters is sent, a blank (0x20) must follow. The blank is ignored in the device.

Example: **Changing limit value 1 (upper limit value for analog input) to 90.5**

Byte	0	1	2	3	4	5
	Func	Limit value	39	30	2E	35
	3	1	'9'	'0'	'.'	'5'

Register	Value (hex)
3216	0301
3217	3930
3218	2E35

Query:

Slave address	05	
Function	10	16: Write Multiple Register
Register	0C 90	Register 3216
No. of registers	00 03	3 registers
No. of bytes	06	6 bytes
Data	03 01 39 30 2E 35	
CRC	3D FE	

Response:

Slave address	05	
Function	10	16: Write Multiple Register
Register	0C 90	Register 3216
No. of registers	00 03	3 registers
CRC	82 F1	

Example: **Changing limit value 3 (gradient for analog input) to 5.7 within 10 seconds**

Byte	0	1	2	3	4	5	6	7
	Func	Limit value	35	2E	37	3B	31	30
	3	3	'5'	'.'	'7'	';'	'1'	'0'

Register	Value (hex)
3216	0303
3217	352E
3218	373B
3219	3130

Query:			
	Slave address	05	
	Function	10	16: Write Multiple Register
	Register	0C 90	Register 3216
	No. of registers	00 04	4 registers
	No. of bytes	08	8 bytes
	Data	03 03 35 2E 37 3B 31 30	
	CRC	94 BF	

Response:			
	Slave address	05	
	Function	10	16: Write Multiple Register
	Register	0C 90	Register 3216
	No. of registers	00 04	4 registers
	CRC	C3 33	

3.2.10.3 Enter the reason for the change in the limit value

Before saving the change in the limit value, it is possible to enter a reason which is saved in the events list. If a reason is not given, the message "Reason: changed by fieldbus" is entered in the events list.

Text items (as per the ASCII table) can be transmitted. The maximum length of the text item is 30 characters. If the text is longer than 30 characters it is truncated and saved.
The text must be written via **16 Write Multiple Register**, 2 characters per register.
If an uneven number of characters is sent, a blank (0x20) must follow. The space is not displayed in the event log.

Byte	0	1
	Func	Limit value
	5	x

Query:			
	Slave address	05	
	Function	10	10: Write Multiple Register
	Register	0C 90	Register 3216
	No. Register	00 07	7 Register
	No. Byte	0E	14 Byte
	Data	05 01	Function 5, Default 1
	Text	52 65 61 73	"Reason why!"
		6F 6E 20 77	
		68 79 21 20	
	CRC	62 64	

Response:			
	Slave address	05	
	Function	10	10: Write Multiple Register
	Register	0C 90	Register 3216
	No. Register	00 07	7 Register
	CRC	83 32	

3.2.10.4 Accepting limit values

Here, the modified limit values are accepted in the device and stored in the device settings.
16 Write Multiple Register or 06 Write Single Register can be used for this purpose.

Byte	0	1
	Func	Fill byte
	2	2A

Register	Value (hex)
3216	022A

Query:	Slave address	05	
	Function	10	16: Write Multiple Register
	Register	0C 90	Register 3216
	No. of registers	00 01	1 register
	No. of bytes	02	2 bytes
	Data	02 2A	
	CRC	C5 7F	
Response:	Slave address	05	
	Function	10	16: Write Multiple Register
	Register	0C 90	Register 3216
	No. of registers	00 01	1 register
	CRC	03 30	

3.2.10.5 Reading out the communication status

This can be used to read out the status of the last limit value function performed.
As a prerequisite, limit value readout should not be enabled (see 3.2.10.6).

Example: **Incorrect function addressed**

Query:	Slave address	05	
	Function	03	03: Read holding register (4x)
	Register	0C 90	Register 3216
	No. of registers	00 01	1 register
	CRC	86 F3	
Response:	Slave address	05	
	Function	3	03: Read holding register (4x)
	No. of bytes	2	2 bytes
	Data	00 01	
	CRC	88 44	

Register	Value (hex)
3216	0001

Communication status:
0: OK
1: Incorrect function number or limit value number
2: Missing data
3: Limit value not active
4: Gradient-> two values
5: Function currently not possible
9: Error

3.2.10.6 Read out limit values

To activate the function, the number of the first desired limit value is transmitted.
By activating this function, reading from Modbus address 3216 no longer returns the communication status.
Instead the limit value settings of the limit value in question is returned in 6 registers.

Byte	0	1
	Func	Limit value
	4	1

Query:	Slave address	05	
	Function	06	06: Write Single Register
	Register	0C 90	Register 3216
	Data	04 01	Function 4, Limit value 1
	CRC	48 33	
Response:	Slave address	05	
	Function	06	06: Write Single Register
	Register	0C 90	Register 3216
	Data	04 01	Function 4, Limit value 1
	CRC	48 33	

Afterwards the desired limit value settings (6 registers) are read out from register 3216.
If the limit value number transmitted is outside the limit value limits (1-100), the following error is indicated in the communication status:

Query:	Slave address	05	
	Function	03	03: Read Holding register (4x)
	Register	0C 90	Register 3216
	No. Register	00 06	6 Register
	CRC	C7 31	

Response:	Slave address	05	
	Function	03	03: Read Holding register (4x)
	No. Bytes	0C	12 Bytes
	Data	00 01	Incorrect limit value number
	Data	00 00 00 00 00 00 00 00 00 00	
	CRC	93 8F	

Otherwise the process of querying the communication status returns the settings of a limit value (see also 3.2.10.7 Tables and definitions):

Response:	Slave address	05	
	Function	03	03: Read Holding register (4x)
	No. Bytes	0C	12 Bytes
	LV,LVType	01 01	Limit value 1, Lower limit value
	Value	40 B0 00 00	5.5
	Span	00 00	Time span for gradient (not required here)
	Delay type	00 00	Seconds
	Delay	00 04	4 Seconds
	CRC	59 9C	

After each query, the limit value number is set to the next activated limit value and can be read out with the next query. After the last activated limit value, the system starts again with the first activated limit value.
If no limit values are activated, all the data are set to 0 in the response.
To deactivate the function, 255 is transmitted as the limit value number or a function not equal to 4 is performed.

3.2.10.7 Tables and definitions

LV: Values between 1 and 100

LVType:

0	Switched off
1	Lower limit value
2	Upper limit value
3-6	Evaluation 1-4
7	Gradient dy/dt
8-11	Evaluation: limit value statistics, frequency
12-15	Evaluation: limit value statistics, duration

Value: Limit value as floating point number (IEEE754, Big Endian)

Span: Time span for gradient (1-60 s)

Delay type: Unit of delay time.

0	Seconds
1	Minutes
2	Hours

Delay: Delay time in the unit set above (0-999).

3.2.11 Modbus master -> Device: transfer text

Text (as per the ASCII table) can be stored in the device's event log. The maximum length of the text item is 40 characters. If it is longer than 40 characters, it is shortened when stored.
The text must be written via **16 Write Multiple Register**, 2 characters per register.
If an odd number of characters is sent, a space must follow (0x20). The space is not displayed in the event log.

Channel	Reg. Dec.	Reg. Hex.	Length in bytes
Text	3024	BD0	Max. 40

Tab. 16: Register addresses for the transfer of text, Modbus master -> Device

Byte	0	1	2	3	4	5
	41	42	43	44	45	20
	‚A'	‚B'	‚C'	‚D'	‚E'	‚ '

Register	Value (hex)
3024	**4142**
3025	**4344**
3026	**4520**

Example: **Generating the text "ABCDE"**

Query:

Slave address	05	
Function	10	16: Write Multiple Register
Register	0B D0	Register 3024
No. of registers	00 03	3 registers
No. of bytes	06	6 bytes
Data	41 42 43 44 45 20	
CRC	D8 4E	

Response:

Slave address	05	
Function	10	16: Write Multiple Register
Register	0B D0	Register 3024
No. of registers	00 03	3 registers
CRC	82 51	

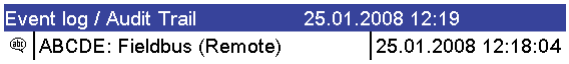


Fig. 14: Entry of text in the event log

3.2.12 Structure of the process values

3.2.12.1 32-bit floating point number (IEEE-754)

Octet	8	7	6	5	4	3	2	1
0	Sign	(E) 2^{7}	(E) 2^{6}					(E) 2^{1}
1	(E) 2^{0}	(M) 2^{-1}	(M) 2^{-2}					(M) 2^{-7}
2	(M) 2^{-8}							(M) 2^{-15}
3	(M) 2^{-16}							(M) 2^{-23}

Sign = 0: Positive number

Sign = 1: Negative number

E = Exponent, M = Mantissa

$$Number = -1^{VZ} \cdot (1 + M) \cdot 2^{E-127}$$

Example: 40 F0 00 00 h = 0100 0000 1111 0000 0000 0000 0000 0000 b

$$\text{Value} = -1^{0} \cdot 2^{129-127} \cdot (1 + 2^{-1} + 2^{-2} + 2^{-3})$$
$$= 1 \cdot 2^{2} \cdot (1 + 0{,}5 + 0{,}25 + 0{,}125)$$
$$= 1 \cdot 4 \cdot 1{,}875 = 7{,}5$$

Byte	0	1	2	3	4	5
	00	80	40	F0	00	00
	Off-limit conditions	Status floating point number	Floating point number = 7.5			

3.2.12.2 Off-limit conditions

Device -> Modbus master

Here the conditions of the first 8 assigned limit values of the channel are registered.

Bit 0: 1st assigned limit value
...
Bit 7: 8th assigned limit value

Bit x = 1: Limit violation
= 0: No limit violation

Example:
If a limit on instantaneous value and a limit value on analysis 1 are assigned to the analog input 1, then the 2 limit value conditions in bit 0 and 1 are indicated in the measured value of analog input 1 (register 256) **and** the integrated analog input 1 (register 528).

Byte	0	1	2	3	4	5
	02	80	40	F0	00	00
	Off-limit conditions	Status floating point number	Floating point number = 7.5			

Bit 0 = 0: No limit violation of the 1st assigned limit value;
here the limit value is set to the instantaneous value.
Bit 1 = 1: Limit violation of the 2nd assigned limit value; here the limit value is set to the integrated value.

3.2.12.3 Status of the floating point number

Device -> Modbus master
10H = e.g. cable open circuit, do not use the value
8xH = value OK
x.bit 0: lower limit value or decreasing gradient
x.bit 1: upper limit value or increasing gradient
x.bit 2: underrange
x.bit 3: overrange

Otherwise = value not OK

Modbus master -> Device
80H: value OK
Not equal to 80H: do not use the value (cable open circuit)

3.2.12.4 Digital status

Modbus master -> Device
The statuses of the 14 digital inputs are transmitted in register 2 (4 bytes) (see Section 3.2.2) also. A digital status is described by two bits. The statuses are found in register 120 and the mask, describing which digital input should adopt the status, in register 121.
The 2 registers must never be written separately, but rather together via **16 Write Multiple Register**.

Register 120 bit x = 0: "Low" status
= 1: "High" status
Register 121 bit x = 0: Do not adopt
= 1: Adopt

Example:

Byte 0 Status (Bit 15-8)	Byte 1 Status (Bit 7-0)	Byte 2 Mask (Bit 15-8)	Byte 3 Mask (Bit 7-0)
XX000000	10000000	XX000001	10000000
Bit 8 Low Digital 9	Bit 7 High Digital 8	Bit 8 High Digital 9 active	Bit 7 High Digital 8 active

Fig. 15: Structure of the 2 registers (4 bytes) transmitted when status is digital (Modbus master -> Device)

Register	Value (hex)
120	0080
121	0180

Fig. 16: Register contents (4 bytes) when status is digital (Modbus master -> Device)

In this case, only bit 7 (digital 8) and bit 8 (digital 9) are adopted (byte 2 and 3).
The statuses for this are bit 8 = low and bit 7 = high (byte 0 and 1).

Device -> Modbus master
The statuses of the 14 digital inputs are transmitted in the first register (high byte bit 0) (see Section 3.2.5 also).

4 List of abbreviations/explanation of terms

Modbus module: The Modbus RTU or Modbus ETH slave plug-in module that is plugged into the rear of the device

Modbus master: All equipment, such as the PLC and PC plug-in boards, that have a Modbus master function

5 Index

fill level | water level | pressure | temperature | flow | visualization | signal converter | sensoric

ACS-CONTROL-SYSTEM
knowhow with system
Your partner for measuring technology and automation

ACS-CONTROL-SYSTEM GmbH
Lauterbachstr. 57
D- 84307 Eggenfelden

Tel.: +49 (0) 8721/ 9668-0
Fax: +49 (0) 8721/ 9668-30

info@acs-controlsystem.de
www.acs-controlsystem.de

Stand 032/2013